ニューラルネットワークの学習による ロバストな制御系の構築法*

中西 弘明**・井上 紘一**

Design Methods of Robust Feedback Controllers by Training Neural Networks*

Hiroaki NAKANISHI** and Koichi INOUE**

This paper proposes two efficient methods to design a robust feedback control system by use of neural networks. The first method is based on L_2 gain, and two different neural networks are used. The controller is trained to be robust as a result of competition between neural networks. The second method is based on MiniMax optimization, and is useful to treat parametric uncertainties. In both methods, robustness of the neural network can be quantified. It is very easy to combine proposed methods so that effective methods for various problems can be derived.

1. はじめに

近年, 非線形な制御対象に対してニューラルネットワークを用いて制御系を構築する様々な方法が提案されている¹⁾⁻³⁾. これらの研究の多くは, 制御対象は完全に既知であるか, 動特性の未知部分をニューラルネットワークを用いて学習させた近似モデルに基づいて学習を行っている. しかし, 従来の研究では学習された制御器のロバスト性に対する考慮がなされていないため, モデルと実対象の間の差がどれぐらいまで許されるのか全く保証されていなかった. ロバスト性を考慮するために, 学習の評価関数に制御対象のパラメータの感度を重み付きで加えて, パラメータの変動が制御結果に大きな影響を与えないように考慮した学習を行うことが考えられる. たとえば, 参考文献 4) では評価関数のパラメータ変数による微分, つまりそのパラメータの感度を評価関数に加えて学習を行う手法を提案しているが, この方法ではどの程度のパラメータ変動までニューラルネットワークは許容できるのか不明であったり, 評価関数に加えるパラメータの感度の重みの決定法に問題がある. また, パラメータが変動したときの制御結果の考慮はされないため, 制

御性能の悪化がどの程度起こるかを学習の段階で保証することはできないうえに, パラメータの不確定さ以外に適用できないことも問題である.

本論文では, ニューラルネットワークを用いて制御対象の不確かさに対してロバストな制御系を学習させる二つの基本的方法を提案する. 第一の方法は L_2 ゲインに基づく方法で, 制御対象の非構造的な不確かさを考慮することを可能にするものである. 本手法では二つのニューラルネットワークを用いる. 一つは制御器として学習を行うネットワークであるが, もう一つは制御系の不確かさを模擬するネットワークである. これらを競合するように学習させることにより, ニューラルネットワークは制御対象の不確かさに対してロバスト性を持った制御器を学習する. また, 本手法では学習の後に得られるニューラルネットワークのロバスト性が定量的に評価されるが, このような定量的なロバスト性の評価は他のニューラルネットワークを制御に用いる研究には見られない優位な点である. 第二の方法は, 制御対象のパラメータ変動を取り扱う方法で, MiniMax 最適化に基づいて学習を行うものである. 学習のときに不確かなパラメータの存在範囲を仮定するので, 学習後のニューラルネットワークがロバスト性を持っている範囲は定量的に保証される. さらに, 本方法で学習を行ったニューラルネットワークの制御性能の最大劣化が学習により分かるので, 安全に用いることができる. 第二の方法は第一の方法に比べると適

* 原稿受付 1998 年 12 月 1 日

** 京都大学 大学院 工学研究科 Graduate School of Engineering, Kyoto University; Yoshida-Honmachi, Sakyo ward, Kyoto city, Kyoto 606-8501, JAPAN

Key Words: neural network, L_2 gain, MiniMax optimization, nonlinear robust control.

用範囲が狭いのは自明であるが、ロバスト制御系によく見られる制御系の保守性の問題が第一の方法に比べて小さくなるという利点がある。これらの提案手法は複合して用いることが可能であり、不確かさの情報（存在範囲や大きさなど）を制御系の学習に効率的に利用する方法へと発展させることが容易である。数値シミュレーションでは、提案する二つの方法の有効性を示す。

2. 対象とする問題

ここでは、本論文が対象とする問題について述べる。制御対象のモデルは、次のような離散時間システムであるとする。

$$\mathbf{x}(t+1) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{v}(t)) \quad (1)$$

(1) 式で $\mathbf{x}(t)$ は n 次元状態変数ベクトル、 $\mathbf{u}(t)$ は m 次元入力ベクトル、 $\mathbf{f}$ は既知である n 次元ベクトル値関数であり、 $\mathbf{v}=0$ のとき制御対象の公称モデルとなる。また、 $\mathbf{v}(t)$ は制御対象の不確かさを表す未知関数である。(1) 式は不確かさの関数構造に対する情報が全くないときの制御対象のモデルであり、存在するあるいは考慮すべき不確かさがパラメータの不確かさに限定される場合には、制御対象のモデルは

$$\mathbf{x}(t+1) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) + \alpha^T \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t)) \quad (2)$$

のように書き直すことができる。(2) 式で $\mathbf{f}$ と $\mathbf{g}$ は既知なベクトル値関数であり、 α は不確かなパラメータを並べたもので、各成分が独立であり、ある領域 Δ 内に存在する時不変なベクトルであるとする。

本論文では上記のような不確かな制御対象に対して制御対象の状態はすべて測定できると仮定し、

$$\mathbf{u}(t) = \mathbf{u}(\mathbf{x}(t)) \quad (3)$$

のようなロバストな状態フィードバック制御器をニューラルネットワークを用いて学習し、構築することを目標とする。また、仮定として、不確かさが存在しても制御系の原点は平衡点であるとする。Fig. 1 にニューラルネットワークを用いたフィードバック制御系のブロック図を示す。本研究では、(1), (2) 式のように不確かさが存在するような対象の原点を安定にするロバストな制御系の構築を目標とする。

3. L_2 ゲインに基づく学習

3.1 ニューラルネットワークによる不確かな系の表現

この手法で取り扱う制御対象は(1)式で表されるものである。この制御対象に対して、評価出力 $z(t)$ を次式の

Fig. 1 Block diagram of feedback controller using neural network

ように定める。

$$z(t) = h(\mathbf{x}(t), \mathbf{u}(t)) \quad (\text{ただし } h(0,0) = 0) \quad (4)$$

(1) 式に含まれている制御対象の不確かさには何ら情報が含まれていないが、不確かさを制御対象のモデル化できていない部分であるとする、状態の関数であるとしてよい。また、不確かさを外乱であるとする、後で述べるように制御系にとって最悪な外乱を考慮することがもつとも安全な考えである。最適制御理論によると、それは制御対象の状態の関数であるので、不確かさを外乱としても状態の関数としてよい。このような不確かさを含む制御対象に対してロバストな制御系をニューラルネットワークを用いて学習させるために Fig. 2 のように異なる二つのニューラルネットワークを配置する。一つは $\mathbf{u}(t)$ を出力する、つまり制御器として用いられるニューラルネットワークである。これに対して、もう一つは $\mathbf{v}(t)$ を出力としており制御対象の不確かさを模擬するために用いられている。本研究では(3)式のような状態フィードバック制御器を学習することが目標であり、さらに不確かさ $\mathbf{v}(t)$ は状態変数 $\mathbf{x}(t)$ の関数としてよいことから、Fig. 2 に含まれるニューラルネットワークとして階層構造のものを用いる。

Fig. 2 Block diagram for training neural networks

3.2 ロバスト制御系の学習

Fig. 2 のシステムに対して次のような評価関数 J を考えて、二つのニューラルネットワークを学習させる。

$$J = \sum_{t=0}^T \|z(t)\|^2 - \gamma^2 \|v(t)\|^2 \quad (5)$$

ここで $\|\cdot\|$ は信号のユークリッドノルムを表す。(5)式で用いられている γ は非負の定数であり、設計パラメータである。もし $\gamma=0$ のように選ぶと、(5)式は最適制御問題の評価関数になることから、この方法は最適制御問題の拡張になっているといえる。ロバスト制御系を学習させるときには $\gamma>0$ のように選ぶ。

制御器として用いるニューラルネットワークは不確かさを含んだ制御系を安定に制御するようにするのが目的であり、(5)式を最小にするように学習を進める。対して、不確かさを模擬しているニューラルネットワークは制御器を含んだシステムにとって最悪の不確かさになるように、つまり (5) 式を最大にするように学習を行う。その結果、ロバスト制御系の学習は次の最適化問題に帰着される。

$$\min_{u(t)} \max_{v(t)} J \quad (6)$$

(5)式に含まれた γ は v から z への L_2 ゲインに相当する。これと小ゲイン定理から、 v のノルムが $1/\gamma$ を越えない範囲までロバスト安定になる^{5),6)}。よって、小さな γ で学習させることは大きなロバスト性を要求していることを意味する。 γ によって学習後のネットワークのロバスト性が定量的に保証される。ロバストな制御系が学習できる γ には明らかに最小値があり、その最小値より小さな γ で学習を行っても、ロバストな制御系を学習することはできない。 γ の最小値を用いて学習を行うと得られる制御系はロバスト性が最大になる。しかし、小さな γ を学習に用いると、得られる制御系のロバスト性が大きくなる反面、保守的になるので γ は慎重に選ぶ必要がある。

(6) 式の最適化は、どちらか一方のニューラルネットワークを固定すると、参考文献 3) で示された評価関数を最適にするフィードバック制御器を学習することに等しい。そこで、(6)式の最適化は一方のネットワークを固定した状態で、他方のネットワークを参考文献 3) に示された方法により学習を行い、その後、固定していたネットワークを学習させるというように、各ネットワークを交互に学習させることで行う。これは、(6)式の変数が制御器としてのネットワークと不確かさを模擬するネットワークの結合度に帰着され、それらは二つに分類することができる。そのために (6) 式の最適化は従来の学習を

応用することで十分に実現することができる。

制御対象に存在する不確かさを制御器にとっての最悪なものと考えておくことがもっとも安全な設計方法であり、この考えは H_∞ 制御⁶⁾などでも使われているものである。ただし、制御器に対する最悪な不確かさは実際にはあり得ない場合もあり、そのような不確かさまで考えてしまうことは制御系の保守性という問題を引き起こす可能性がある。

3.3 ロバスト性の定量化

ロバストな制御系を構築するだけでなく、制御系のロバスト性を定量的に測定することも非常に重要である。上記の方法を少し変更するだけで制御器のロバスト性の定量的な測定を行うことが可能である。先に述べたロバスト制御系の学習では (6) 式のように学習を行うが、制御器を固定して (7) 式のように不確かさを模擬するネットワークだけを学習させる。

$$\max_{v(t)} J \quad (7)$$

このとき、制御系が安定である範囲で γ を減らしていき、 γ の最小値を繰り返し計算により求めると、それが与えられた制御系のロバスト性の定量的な指標になる。

最適フィードバック制御系を学習させたニューラルネットワーク³⁾のロバスト性の測定はこの方法により初めて可能になる。また、本手法でロバスト性の定量化できる制御器はニューラルネットワークで実現されているものだけでなく任意のものである。

4. MiniMax 最適化に基づく学習

4.1 ロバスト制御系の学習

もし不確かさに対する情報があつたとしても、3.で説明した方法ではそれを有効に利用できない。このため、先験的な不確かさの情報を有効利用することが可能な学習方法も必要である。不確かさに関する最も詳しい先験的情報とは、制御対象に含まれる不確かさはパラメータの不確定さであると分かっていることである。このときの制御対象は (2) 式のように表される。このような場合、3.で不確かさを扱った方法をとると、実際に存在し得ない、あるいは考慮しなくてもよい不確かさまでが考慮されるために、得られる制御系が保守的になってしまうことは避けられない。このような不都合を解消するために、パラメータの不確かさを考慮するロバストな制御系の学習法が必要である。しかし、ロバスト性の考慮のためにパラメータの感度関数を評価関数に加えて学習を行うことはパラメータの変動に対してロバストな領域を学習時には明らかでないという難点がある。

仮定より、(2)式に含まれている不確かなパラメータベ

クトル α は公称値パラメータ α_{nom} を含む閉集合 Δ 内に存在する。 α_{nom} に対して、 L を正定値関数として、

$$J_0 = \sum_{t=0}^T L(x(t), u(t)) \quad (8)$$

のような評価関数を考えて最適フィードバック制御系をニューラルネットワークに学習させることは、パラメータ α の変動を考慮しておらず、パラメータが公称値以外のときの制御結果を保証できない。 Δ 内のどこに α が存在しても制御対象を安定にする制御系が必要である。このようなロバストな制御系の学習のために、次のような評価関数 J を考える。

$$J = \max_{\alpha \in \Delta} J_0 \quad (9)$$

この評価関数 J を最小にするように Fig. 1 のブロック図のように構成したニューラルネットワークを学習させる。 (9) 式を用いて学習を行うことは、領域 Δ 内で最も制御結果を悪くするパラメータ、つまり制御器にとって最悪パラメータ下にある制御対象の制御評価を最適にすることである。このように学習を行ったニューラルネットワークでは学習後の評価関数が領域 Δ 内の最悪の制御評価となり、最悪の変動が起こった場合の制御性能が定量的に保証されているロバストな制御器となる。

3. で述べたように、ここでも最悪の場合を考えて学習を行っている。しかし、 L_2 ゲインに基づく学習と異なり、考慮される不確かさには先験的情報が加わっているために、制御性能をそれほど劣化させずにロバスト性を考慮できる。領域 Δ の大きさが制御性能の劣化に大きな影響を与える。 (9) 式では Δ に含まれるすべてのパラメータを考慮しているが、実際には最悪なパラメータは Δ の頂点と考えてよいので、 $V(\Delta)$ を Δ のすべての頂点として、

$$J = \max_{\alpha \in V(\Delta)} J_0 \quad (10)$$

と省略化して学習を行う。しかし、 (9), (10) 式のいずれも微分不可能な評価関数になっており、勾配法^{2),3),7)}などの学習アルゴリズムを使うことができない。

4.2 Powell の共役方向法による学習アルゴリズム

4.1 で述べたように、 (9), (10) 式に基づいて学習を行うには勾配法では不十分である。そこで、評価関数の微係数を必要としない Powell の共役方向法による学習アルゴリズム^{8),9)}により学習を行う³⁾。

従来から用いられている勾配法と比較して、この学習アルゴリズムには微分可能でない評価関数を用いることができるという優位性があるだけでなく、実行に必要な

情報が探索点における評価関数の大きさだけであり、勾配法では不可欠な情報であったプラントのヤコビアンが必要でなくなる。このために、実対象を繰り返し使って学習させたり、内部情報（たとえば状態方程式やそれに含まれるパラメータ）を隠した制御対象のシミュレータなどからも学習を行うことができるなどの適用できる範囲が広い。 L_2 ゲインに基づく学習の場合は、勾配法を用いても学習を行うことができるが、本アルゴリズムにより学習を行うとプラントのヤコビアンを求めることが不要となるため、学習の時間の短縮に大きく貢献するので、後に示す数値シミュレーションでは本アルゴリズムを用いて学習を実行させている。

5. 数値シミュレーション

本論文で提案した手法を適用した数値計算の例を示す。数値例では、用いたニューラルネットワークの中間層のユニット数はいずれも 10 個とした。

5.1 L_2 ゲインに基づく学習の例

ここでは 3. で説明した手法で学習した例を示す。学習は、一方のネットワークが 20 回共役方向法で学習を行ったのち、他方のネットワークが 20 回学習を行うという繰り返しを 20 回行うとした。制御対象の方程式を以下の通りとする。 h は 0.01 とした。

$$\begin{bmatrix} x_1(t+1) \\ x_2(t+1) \end{bmatrix} = \begin{bmatrix} x_1(t) + hx_2(t) \\ x_2(t) + h\{ (1-x_1^2(t))x_2(t) \\ -x_1(t) + u(t) + v(t) \} \end{bmatrix} \quad (11)$$

$$z(t) = [x_1(t), x_2(t), u(t)]^T \quad (12)$$

また、 $(x_1(0), x_2(0)) = (0.5, 0)$ とした。評価関数を次式で定める。

$$J = \sum_{t=0}^{1000} [\|z(t)\|^2 - \gamma^2 v^2(t)] \quad (13)$$

比較検討のために (11) 式で $v(t) \equiv 0$ と仮定し、評価関数 (13) 式で $\gamma = 0$ とし最適フィードバック制御系を学習させたニューラルネットワークを用いる。 3. で説明したロバスト性を定量的に測定する方法をこのネットワークに対して適用したところ、 $\gamma = 2.832$ を得た。これに対して、 (13) 式で $\gamma = 1.549$ としロバスト制御系をニューラルネットワークに学習させた。また、この γ の値はロバスト制御系を学習できる最小値であった。つまり、学習で得られた制御系はロバスト性が最大になっている。 γ の値から、この制御系は最適フィードバック制御系を学習したものに比べて、およそ 2 倍のロバスト性を持っていることが分かる。不確かさが存在していないとした制御対象を上記のように学習した二つのニューラルネットワー

Fig. 3 Responses of the nominal plant

クを用いて制御を行った結果を Fig. 3 に示す。この図を見れば分かるように、ロバスト制御系を学習したニューラルネットワークは最適フィードバック制御系を学習したものよりも応答が遅く、制御性能が劣っている。ロバスト制御系の学習の際に、もっと大きな γ を用いて学習すればこのような制御性能の劣化は少なくなる。つぎに、大きな不確かさが存在するとした場合のシミュレーションを行った結果を Fig. 4 に示す。ここで、不確かさのために状態方程式 ($x_2(t)$ に関する式) が次のように変化してしまつたと仮定した。

$$x_2(t+1) = x_2(t) + h \left\{ 1.5(1 - x_1^2(t))x_2(t) - 1.5x_1(t) + 0.5u(t) \right\} \quad (14)$$

この例から分かるように、不確かさの存在のために最適フィードバック制御系は制御対象を制御できていないが、本方式で学習したニューラルネットワークを用いた制御系であれば不確かさが存在するときでも安定に制御ができていることが分かる。 γ の値の違いによって、

Fig. 4 Responses of the plant with big uncertainty

提案した手法により学習したネットワークのロバスト性が高いことが示されていたが、この結果はそのことを裏付ける結果である。

5.2 MiniMax 最適化に基づく学習の例

ここでは 4. で説明した手法で学習した例を示す。制御対象の方程式を以下の通りとする。 h は 0.01 とした。

$$\begin{bmatrix} x_1(t+1) \\ x_2(t+1) \\ x_3(t+1) \\ x_4(t+1) \end{bmatrix} = \begin{bmatrix} x_1(t) + h \cdot x_2(t) \\ x_2(t) + h(-G \cdot \sin x_3(t) + x_1 x_4(t)^2) \\ x_3(t) + h \cdot x_4(t) \\ x_4(t) + h \cdot I \cdot u(t) \end{bmatrix} \quad (15)$$

$$|u| \leq 1.0, \quad x(0) = (2, 0, 0, 0) \quad (16)$$

この制御対象には不確定パラメータ G, I が存在している。 G は公称値が 9.8 で $\pm 10\%$ の誤差を、 I は公称値が 1.0 で $\pm 20\%$ の誤差を含んでいるとする。

$$G \in [8.82, 10.78], \quad I \in [0.8, 1.2] \quad (17)$$

学習に用いる評価関数 (9), (10) 式に含まれる J_0 は次のように定めた。このように評価関数を定めると対象の応答の早い制御器を要求することになるが、一般にロバスト性と制御性能は相反する要求である。

$$J_0 = 0.5 \sum_{t=0}^{2000} \left\{ 100x_1(t)^2 + x_2(t)^2 + x_3(t)^2 + x_4(t)^2 + u^2(t) \right\} \quad (18)$$

5.1 の例と同様に、公称パラメータに対して (18) 式を評価関数として最適フィードバック制御系を学習させたニューラルネットワークと本手法により学習させたネットワークを比較する。これらの学習は、共役方向法を繰り返して 40 回行うこととした。この場合、最適フィードバック制御系を学習させたニューラルネットワークはパラメータ変動に対して非常に弱く、わずか 1% のパラメータ変動すらも許容できず、安定に制御を行うことができない場合があった。しかし、本手法で学習させたニューラルネットワークは (17) 式で表されたパラメータ全領域を安定に制御できた。Fig. 5 に公称パラメータの場合の応答を示す。この図を見れば分かるように、公称パラメータの場合であっても、本手法で学習したニューラルネットワークは最適フィードバック制御系を学習したものとは比べてそれほど制御性能が劣化せず良好な結果を得ている。5.1 の例ではロバスト性の向上に伴って制御性能の劣化が見られたが、不確かさの先験的情報を利用する本手法で

Fig. 5 Responses of the nominal plant

Fig. 6 Responses under the worst parameters

学習すると制御性能をあまり落とさないロバスト制御系が学習できることが分かる。Fig. 6 に本手法で学習したニューラルネットワークにとって最悪なパラメータの場合の応答を示した。最適フィードバック制御を学習したものはまったく制御できていないが、本手法によるものは、最悪のパラメータ下であっても安定に制御できている。さらに、提案手法で学習したネットワークの(18)式に含まれるさまざまなパラメータでの評価関数の値(ただし $\log_{10}$ をとった)を Fig. 7 に示す。Fig. 7 の中央が公称パラメータであるが、(18)式で示されたパラメータの存在領域全体で評価関数が大きくは変化していないことが分かる。よって、提案手法により学習されたニューラルネットワークは、不確定パラメータ G, I の変化に対してロバストな制御系となっていることが分かる。

6. おわりに

本論文では、ニューラルネットワークを用いてロバストな制御系を学習する基本的方法を二つ提案し、それらの有効性を確認した。提案した方法は、学習後のネットワークのロバスト性に対する評価が定量的に行える。ま

Fig. 7 Performance index in Δ (proposed method)

た、任意の制御系のロバスト性を定量的に評価することもニューラルネットワークを学習させることにより可能であることを示した。

L_2 ゲインに基づく学習法は不確かさに対する先験的情報がまったく無い場合に有効であるが、学習で得られる制御系は保守的となることがある。不確かさの先験的情報の典型な例であるパラメータの不確定を伴う制御対象に対しては、MiniMax 最適化による学習法で得られる制御系のほうが制御性能が好ましいものとなる。提案するこれらの方法は適切なロバスト制御系をニューラルネットワークを用いて学習するために基本となる方法であり、複合して用いることにより適切な形で不確かさに関する先験的情報を制御系の学習に組み込むことができ、様々な応用が考えられる。ロバスト制御系の学習ならびにロバスト性の評価で最小の γ を求める手法は現在のところ、効率的な方法が存在せず、繰り返し計算を行って定めている。いずれの場合でも、 γ を徐々に小さくしていくと、ある値を境に解の分岐のような現象が確認され、評価関数の大きさが突然大きくなり、対象は不安定になるので γ の判定は容易であった。しかし、最小の γ を求めることは実用上多くないと思われる。それは、ある γ に定めて提案手法で学習された制御系はその γ がロバスト性能の最小値を示しており、制御系設計においては安全側にマージンがある。小さな γ を用いて学習させると、制御系の保守性の問題を引き起すために実用上はそんなに好ましい制御系にならないことが多い。ロバスト性の評価の場合であっても、ある γ で本手法で安定であると判断されるときもそのロバスト性は安全側にマージンを持って評価される。最小の γ を評価したいときも多いと思われるが、それは繰り返し計算に頼らざるを得ない。しかし、Powell の共役方向法を学習アルゴリズムとすると高速に学習が進む。実際に本論文で用いた数値例は学習回数がわずかに数回でロバスト性が評価できた。ま

た、この所要時間は Pentium II-400MHz の PC で 3 分程度であった。ネットワークの初期点による局所解を防ぐために、複数の初期点から学習を行うとしても、これは十分に実用に耐えるものであろう。ロバスト制御系の学習は、やはり最適フィードバック制御系の学習に比べて時間を必要とするが、Powell の共役方向法を学習アルゴリズムとして採用することによりプラントのヤコビアン計算を省けることなど高速化を意図して行った。実際に行った計算では十分な回数を繰り返しているため、6 時間ほど時間を必要としたが、実際には数回の繰り返して学習が進むことが多く、より短い時間で学習が進んでしまうことが確認された。

ニューラルネットワークは局所解に陥ってしまうことがあるが、それは制御系のネットワークの局所解より不確かさのネットワークの局所解の方が重大な問題である。つまり L_2 ゲインに基づく学習の方に大きな影響を与える。ロバスト性を誤って高く評価してしまう可能性を持つからである。このために、本研究で行った数値シミュレーションでは不確かさを模擬するネットワークを複数の初期点から学習させ、これを防いだ。探索空間を多様な探索を行う学習手法が必要であると考えられるが、この件に関しては今後の課題といえる。MiniMax 最適化による学習の方では、ネットワークを何度か変えて試行するだけで十分である。このときも、共役方向法に基づく学習アルゴリズムが高速であるので簡単に判別できた。

参考文献 10) により、階層構造型ニューラルネットワークは任意の精度で任意の非線形関数を近似できることが知られている。よって、本論文でニューラルネットワークを用いて学習・構築される制御器はネットワークへの入力に関して連続なものに限られる。不確かさを模擬するネットワークは、制御系の最悪な不確かさを十分に表現できる能力を要求されるが、これは参考文献 10) に示された結果により保証される。一方、制御器として用いるニューラルネットワークでは、その構造がロバスト性に大きな影響を与えることが、最近の報告参考文献 11) により明らかとなった。その報告には、中間層に含まれるユニット数がある数以上であれば目的とする制御器を模擬できるが、そのロバスト性は中間層のユニット数により大きく影響されることが示されている。その結果は、制御器として用いられたニューラルネットワークの汎化能力がロバスト性に重大な影響を与えることを示しており、制御器としてニューラルネットワークを用いる場合の汎化能力の定量的な指標としてロバスト性を用いることが有用であることを示している。

数値シミュレーションで示したように、初期値を固定して学習を行っているが、この影響はやはり存在する。初期

条件による影響は参考文献 3) ではニューラルネットワークの汎化能力に依存していたが、本論文で提案した手法ではその影響は緩和される。たとえば、 L_2 ゲインに基づく学習方法では不確かさを模擬するネットワークが存在するため状態フィードバック制御器の学習中に非常に多くの初期点を考慮しているような効果が現われる。それにもかかわらず、提案した方法を単独で用いるときには、初期条件への依存が多少存在してしまう。この影響を完全に考慮することは、本論文で提案した二つの手法を複合化することで、ある領域内の最悪の初期条件を考慮するようなニューラルネットワークの学習方法が導くことにより可能となる。初期条件以外でも、時変パラメータの存在するロバスト制御系や、スライディングモード制御系などさまざまな制御系の設計は提案手法の応用から導かれる。また、同様の枠組みでロバストな状態推定器や動的な制御器を学習する方法を導くことができると考えられるが、それらは今後の課題である。

参考文献

- 1) K. S. Narendra and K. Parthasarathy: Identification and control of dynamical systems using neural networks; *IEEE Trans. Neural Networks*, Vol. 1, No. 1, pp. 4-27 (1990)
- 2) T. R. Niesler and J. J. du Plessis: Time-optimal control by means of neural network; *IEEE Control Systems*, Vol. 15, No. 5, pp. 23-33 (1995)
- 3) 中西, 幸田, 井上: ニューラルネットワークを用いた最適フィードバック制御系の構成法; 計測自動制御学会論文集, Vol. 33, No. 9, pp. 882-889 (1997)
- 4) 大林, 平澤, 須納瀬: 一般化ネットワークにおけるハイブリッド制御方式; 計測自動制御学会第 36 回学術講演会予稿集, Vol. 2, pp. 745-746 (1997)
- 5) 島, 石動, 山下ほか: 非線形システム制御論, コロナ社 (1997)
- 6) J. C. Doyle, K. Glover, P. P. Khargonekar and B. A. Francis: State-space solutions to standard H_2 and H_∞ control problems; *IEEE Trans. Automatic Control*, Vol. 34, No. 8, pp. 831-847 (1989)
- 7) K. S. Narendra and K. Parthasarathy: Gradient methods for the optimization of dynamical systems containing neural networks; *IEEE Trans. Neural Networks*, Vol. 2, No. 4, pp. 252-262 (1991)
- 8) M. J. D. Powell: An efficient method for finding the minimum of a function of several variables without calculating derivatives; *Computer Journal*, Vol. 7, pp. 155-162 (1964)
- 9) 坂和: 最適化と最適制御, 森北出版 (1980)
- 10) K. Funahashi: On the approximate realization of continuous mappings by neural network; *Neural Networks*, Vol. 2, No. 3, pp. 183-192 (1989)
- 11) 中西, 井上: ニューラルネットワークを用いたロバストなフィードバック制御系の学習法; 第 8 回インテリジェント・システム・シンポジウム講演論文集, pp. 523-528 (1998)